SphereFace:

Deep Hypersphere Embedding for Face Recognition

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, Le Song

Georgia Institute of Technology, Carnegie Mellon University, Sun Yat-Sen University

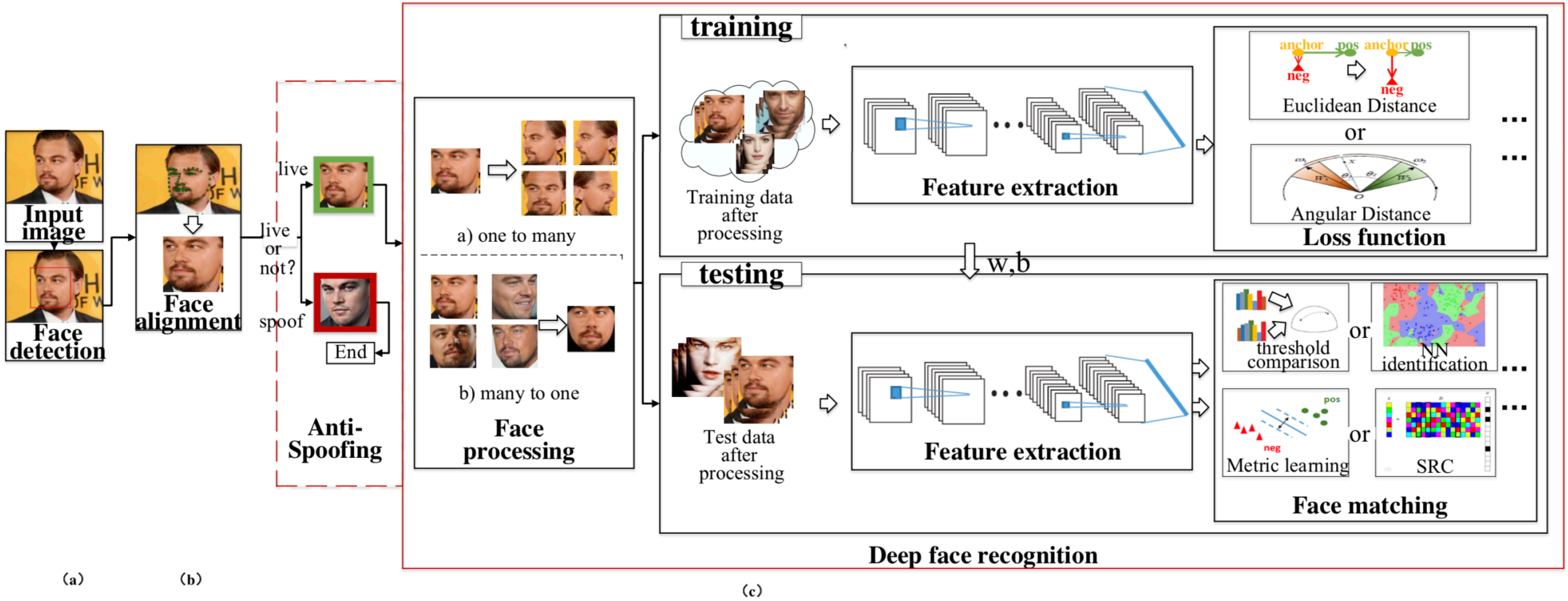
CVPR 2017

Sungman, Cho.

Introduction

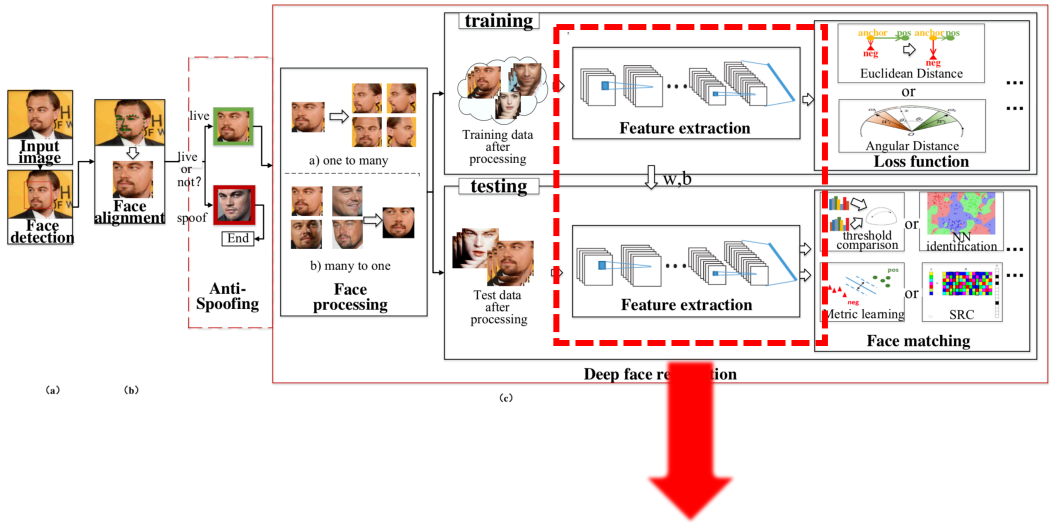
Dive into FR(Face Recognition)

Deep FR(Face Recognition) System



Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

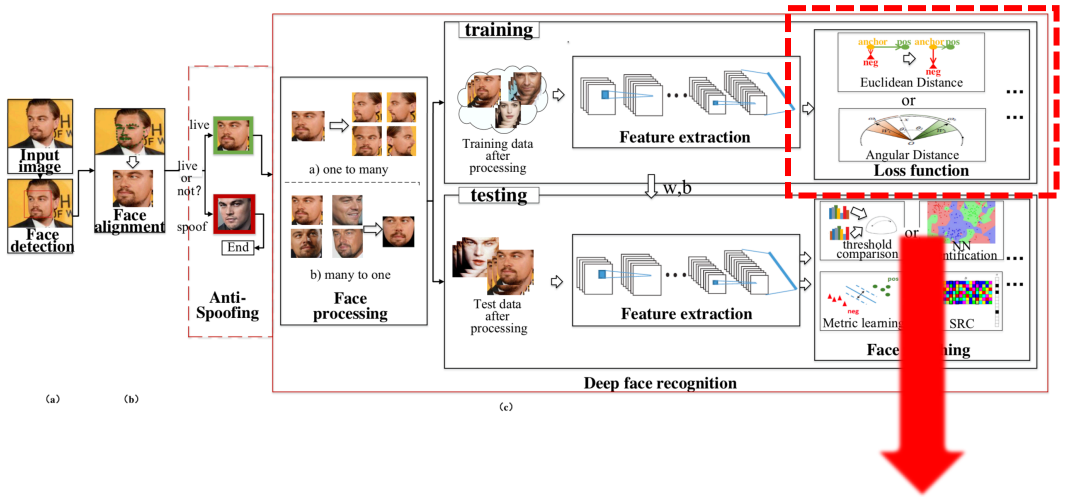
Feature Extraction Network



Network Architectures	Subsettings
backbone network	mainstream architectures: AlexNet [140], [139], [144], VGGNet [123], [116], [224], GoogleNet [204], [144], ResNet [106], [224], SENet [20]
	special architectures [187], [188], [157], [34], [194]
	joint alignment-representation architectures [64], [186], [237], [29]
multiple networks	multipose [87], [115], [211], [175], multipatch [105], [239], [46], [155], [156], [152] [185], multitask [131]

Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

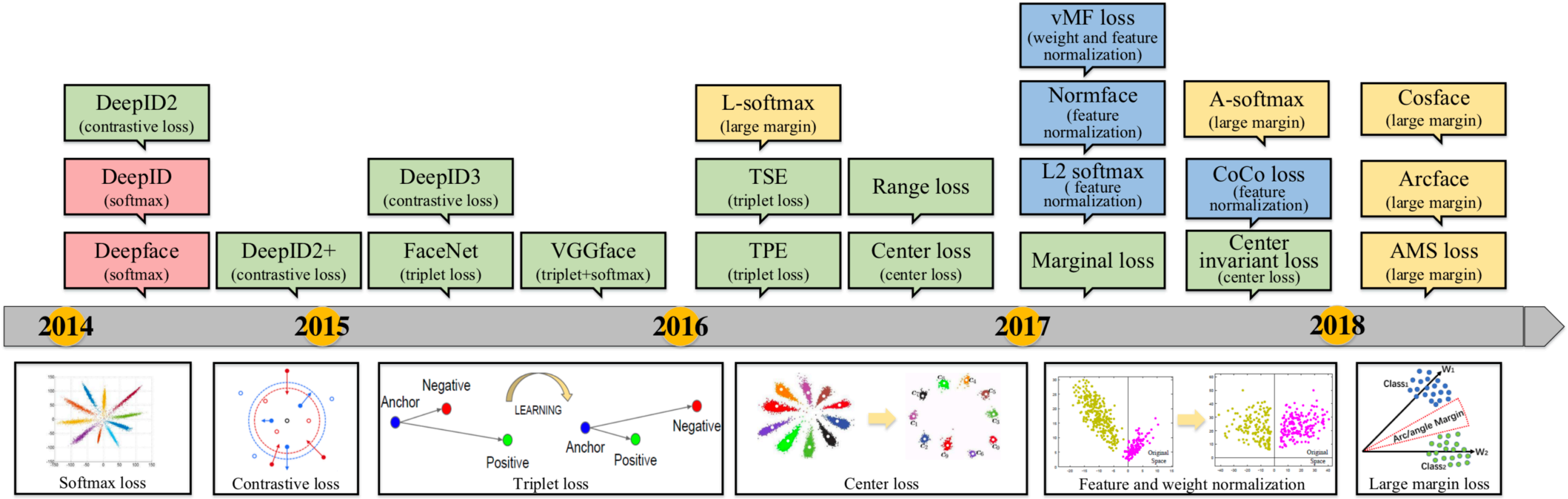
Loss Function



Loss Functions	Brief Description
Euclidean-distance-based loss	compressing intra-variance and enlarging inter-variance based on Euclidean distance. [152], [185], [153], [181], [191], [224], [144], [123], [140], [139], [105], [28]
angular/cosine-margin-based loss	making learned features potentially separable with larger angular/cosine distance. [107], [106], [170], [38], [172], [108]
softmax loss and its variations	modifying the softmax loss to improve performance. [129], [171], [61], [111] [128], [23], [62]

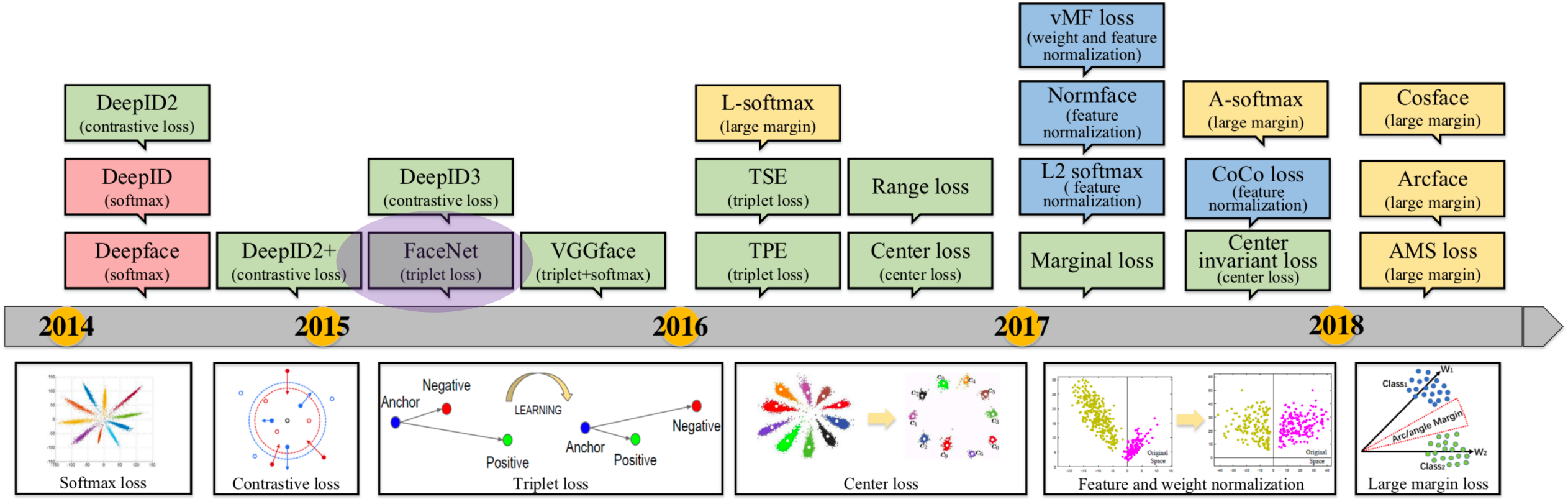
Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Loss Function



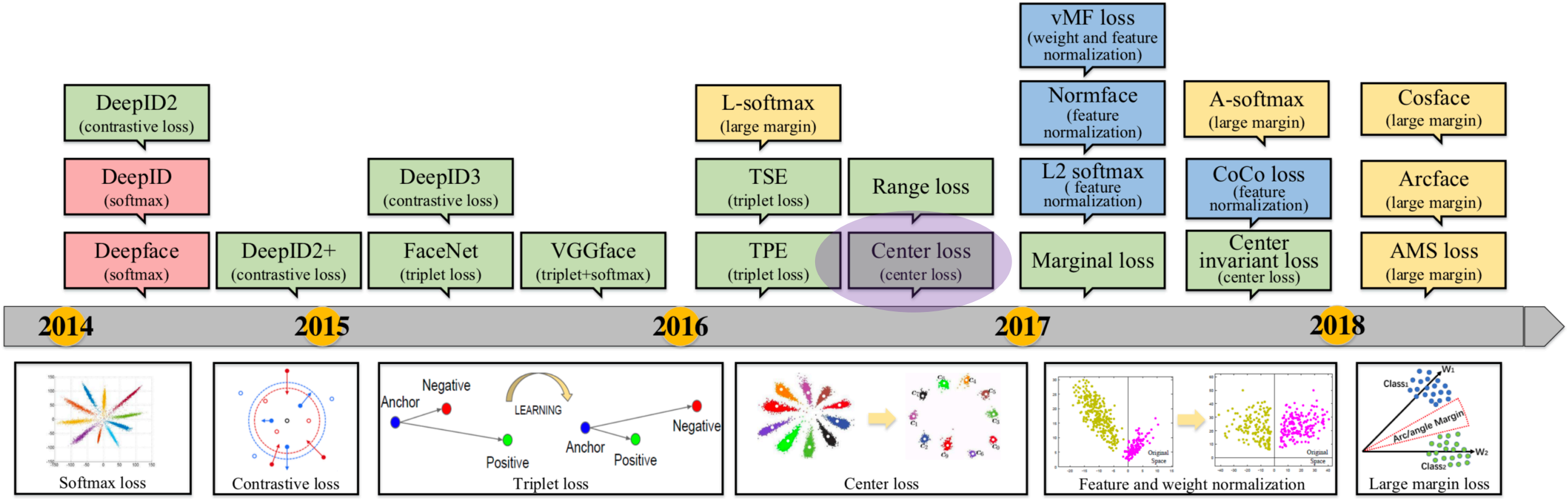
Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Loss Function



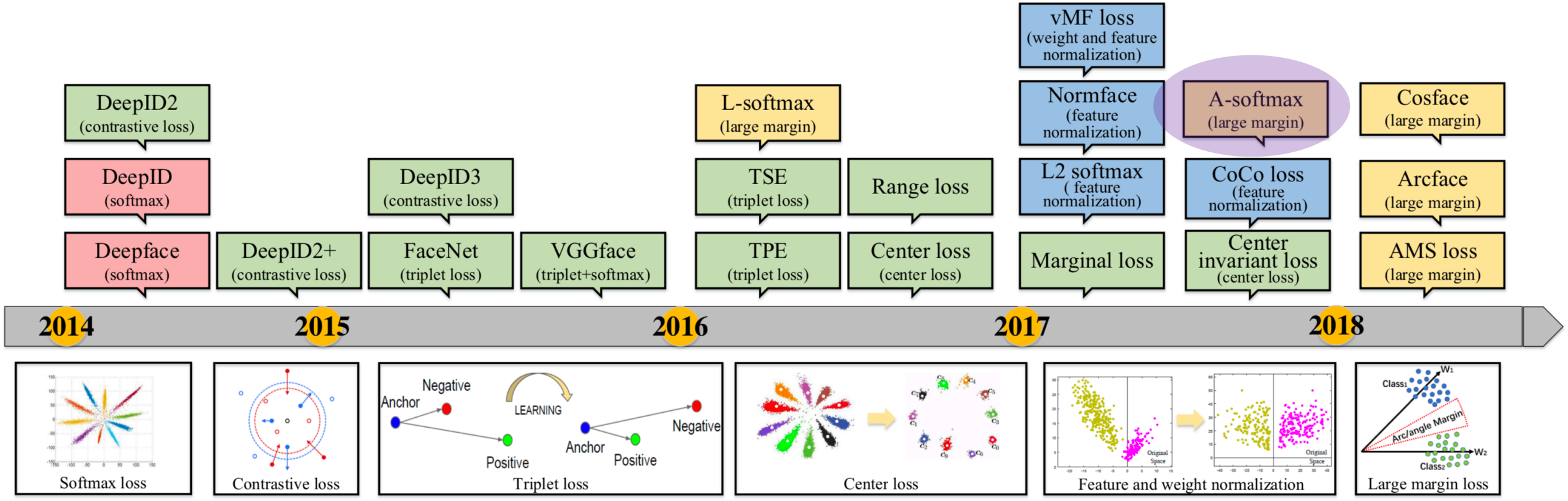
Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Loss Function



Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Loss Function



Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Main stream of FR

- **Softmax**

  Train a multi-class classifier which can separate different identities in the training set

- **Triplet loss (CVPR, 2015)**

  Learn directly an embedding

Main stream of FR

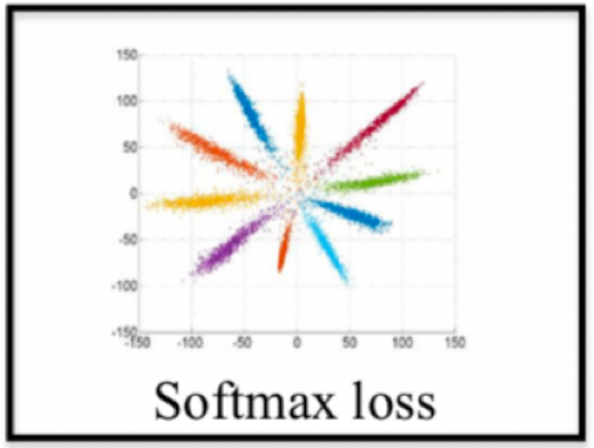
- **Softmax**

  Train a multi-class classifier which can separate different identities in the training set

- **Triplet loss (CVPR, 2015)**

  Learn directly an embedding

Main stream of FR



- **Softmax**

Train a multi-class classifier which can separate different identities in the training set

[cons]

- The size of the linear transformation matrix $W \in \mathbb{R}^{d \times n}$ increases linearly.
- The learned features are separable for the closed-set classification problem. (not discriminative enough for the open-set face recognition problem.)

Main stream of FR

- **Softmax**

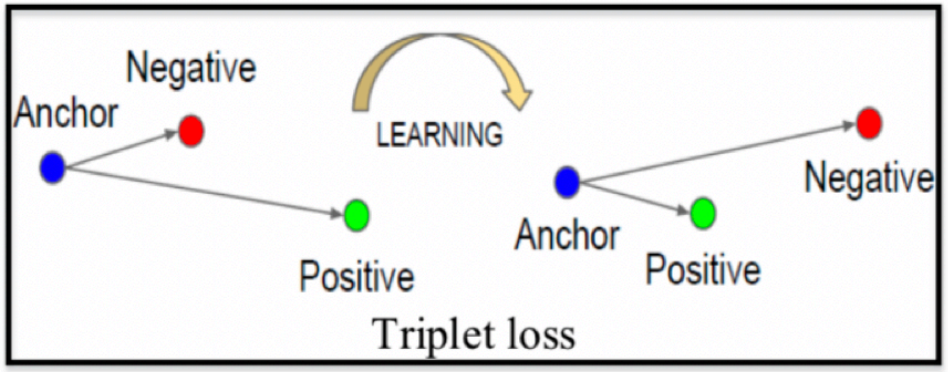
  Train a multi-class classifier which can separate different identities in the training set

- **Triplet loss (CVPR, 2015)**

  Learn directly an embedding

Main stream of FR

- **Triplet loss (CVPR, 2015)**



Learn directly an embedding

[cons]

- **Combinatorial explosion in the number of face triplets.**
- Semi-hard sample mining is a quite difficult problem for effective training.

Variants

- **Center loss (ECCV, 2016)**
- **Sphereface (CVPR, 2017)**
- **ArcFace (arXiv:1801.07698)**

Method	Public. Time	Loss	Architecture	Number of Networks	Training Set	Accuracy±Std(%)
DeepFace [160]	2014	softmax	Alexnet	3	Facebook (4.4M,4K)	97.35±0.25
DeepID2 [152]	2014	contrastive loss	Alexnet	25	CelebFaces+ (0.2M,10K)	99.15±0.13
DeepID3 [153]	2015	contrastive loss	VGGNet-10	50	CelebFaces+ (0.2M,10K)	99.53±0.10
FaceNet [144]	2015	triplet loss	GoogleNet-24	1	Google (500M,10M)	99.63±0.09
Baidu [105]	2015	triplet loss	CNN-9	10	Baidu (1.2M,18K)	99.77
VGGface [123]	2015	triplet loss	VGGNet-16	1	VGGface (2.6M,2.6K)	98.95
light-CNN [188]	2015	softmax	light CNN	1	MS-Celeb-1M (8.4M,100K)	98.8
Center Loss [181]	2016	center loss	Lenet+-7	1	CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K)	99.28
L-softmax [107]	2016	L-softmax	VGGNet-18	1	CASIA-WebFace (0.49M,10K)	98.71
Range Loss [224]	2016	range loss	VGGNet-16	1	MS-Celeb-1M, CASIA-WebFace (5M,100K)	99.52
L2-softmax [129]	2017	L2-softmax	ResNet-101	1	MS-Celeb-1M (3.7M,58K)	99.78
Normface [171]	2017	contrastive loss	ResNet-28	1	CASIA-WebFace (0.49M,10K)	99.19
CoCo loss [111]	2017	CoCo loss	-	1	MS-Celeb-1M (3M,80K)	99.86
vMF loss [62]	2017	vMF loss	ResNet-27	1	MS-Celeb-1M (4.6M,60K)	99.58
Marginal Loss [39]	2017	marginal loss	ResNet-27	1	MS-Celeb-1M (4M,80K)	99.48
SphereFace [106]	2017	A-softmax	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.42
CCL [128]	2018	center invariant loss	ResNet-27	1	CASIA-WebFace (0.49M,10K)	99.12
AMS loss [170]	2018	AMS loss	ResNet-20	1	CASIA-WebFace (0.49M,10K)	99.12
Cosface [172]	2018	cosface	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.33
Arcface [38]	2018	arcface	ResNet-100	1	MS-Celeb-1M (3.8M,85K)	99.83
Ring loss [235]	2018	Ring loss	ResNet-64	1	MS-Celeb-1M (3.5M,31K)	99.50

Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Variants

- **Center loss (ECCV, 2016)**
- Sphereface (CVPR, 2017)
- ArcFace (arXiv:1801.07698)

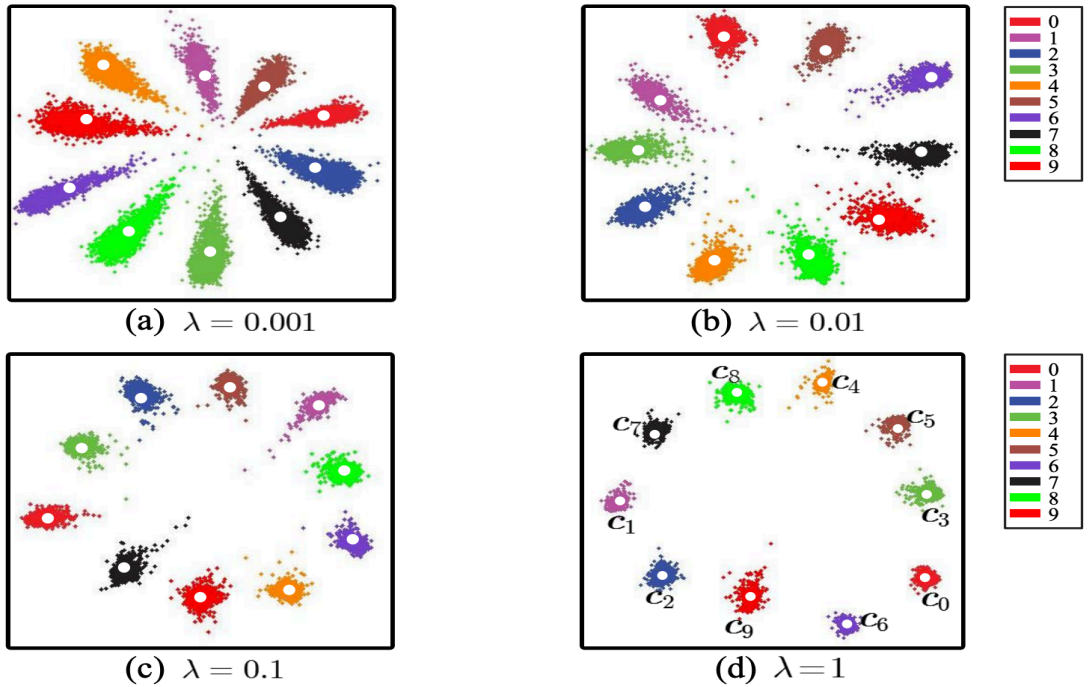
Method	Public. Time	Loss	Architecture	Number of Networks	Training Set	Accuracy±Std(%)
DeepFace [160]	2014	softmax	Alexnet	3	Facebook (4.4M,4K)	97.35±0.25
DeepID2 [152]	2014	contrastive loss	Alexnet	25	CelebFaces+ (0.2M,10K)	99.15±0.13
DeepID3 [153]	2015	contrastive loss	VGGNet-10	50	CelebFaces+ (0.2M,10K)	99.53±0.10
FaceNet [144]	2015	triplet loss	GoogleNet-24	1	Google (500M,10M)	99.63±0.09
Baidu [105]	2015	triplet loss	CNN-9	10	Baidu (1.2M,18K)	99.77
VGGface [123]	2015	triplet loss	VGGNet-16	1	VGGface (2.6M,2.6K)	98.95
light-CNN [188]	2015	softmax	light CNN	1	MS-Celeb-1M (8.4M,100K)	98.8
Center Loss [181]	2016	center loss	Lenet+-7	1	CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K)	99.28
L-softmax [107]	2016	L-softmax	VGGNet-18	1	CASIA-WebFace (0.49M,10K)	98.71
Range Loss [224]	2016	range loss	VGGNet-16	1	MS-Celeb-1M, CASIA-WebFace (5M,100K)	99.52
L2-softmax [129]	2017	L2-softmax	ResNet-101	1	MS-Celeb-1M (3.7M,58K)	99.78
Normface [171]	2017	contrastive loss	ResNet-28	1	CASIA-WebFace (0.49M,10K)	99.19
CoCo loss [111]	2017	CoCo loss	-	1	MS-Celeb-1M (3M,80K)	99.86
vMF loss [62]	2017	vMF loss	ResNet-27	1	MS-Celeb-1M (4.6M,60K)	99.58
Marginal Loss [39]	2017	marginal loss	ResNet-27	1	MS-Celeb-1M (4M,80K)	99.48
SphereFace [106]	2017	A-softmax	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.42
CCL [128]	2018	center invariant loss	ResNet-27	1	CASIA-WebFace (0.49M,10K)	99.12
AMS loss [170]	2018	AMS loss	ResNet-20	1	CASIA-WebFace (0.49M,10K)	99.12
Cosface [172]	2018	cosface	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.33
Arcface [38]	2018	arcface	ResNet-100	1	MS-Celeb-1M (3.8M,85K)	99.83
Ring loss [235]	2018	Ring loss	ResNet-64	1	MS-Celeb-1M (3.5M,31K)	99.50

Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Center Loss

- The Euclidean distance between each **feature vector and its class center**
- To obtain intra-class compactness & inter-class dispersion
- Updating the actual centers during training is difficult.

  (the number of face classes has dramatically increased)



$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$

$$= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \| \boldsymbol{x}_i - \boldsymbol{c}_{y_i} \|_2^2$$

Ref. Y.Wen, "A Discriminative Feature Learning Approach for Deep Face Recognition", ECCV, 2016.

Variants

- Center loss (ECCV, 2016)
- **Sphereface (CVPR, 2017)**
- ArcFace (CVPR, 2019)

Method	Public. Time	Loss	Architecture	Number of Networks	Training Set	Accuracy±Std(%)
DeepFace [160]	2014	softmax	Alexnet	3	Facebook (4.4M,4K)	97.35±0.25
DeepID2 [152]	2014	contrastive loss	Alexnet	25	CelebFaces+ (0.2M,10K)	99.15±0.13
DeepID3 [153]	2015	contrastive loss	VGGNet-10	50	CelebFaces+ (0.2M,10K)	99.53±0.10
FaceNet [144]	2015	triplet loss	GoogleNet-24	1	Google (500M,10M)	99.63±0.09
Baidu [105]	2015	triplet loss	CNN-9	10	Baidu (1.2M,18K)	99.77
VGGface [123]	2015	triplet loss	VGGNet-16	1	VGGface (2.6M,2.6K)	98.95
light-CNN [188]	2015	softmax	light CNN	1	MS-Celeb-1M (8.4M,100K)	98.8
Center Loss [181]	2016	center loss	Lenet+-7	1	CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K)	99.28
L-softmax [107]	2016	L-softmax	VGGNet-18	1	CASIA-WebFace (0.49M,10K)	98.71
Range Loss [224]	2016	range loss	VGGNet-16	1	MS-Celeb-1M, CASIA-WebFace (5M,100K)	99.52
L2-softmax [129]	2017	L2-softmax	ResNet-101	1	MS-Celeb-1M (3.7M,58K)	99.78
Normface [171]	2017	contrastive loss	ResNet-28	1	CASIA-WebFace (0.49M,10K)	99.19
CoCo loss [111]	2017	CoCo loss	-	1	MS-Celeb-1M (3M,80K)	99.86
vMF loss [62]	2017	vMF loss	ResNet-27	1	MS-Celeb-1M (4.6M,60K)	99.58
Marginal Loss [39]	2017	marginal loss	ResNet-27	1	MS-Celeb-1M (4M,80K)	99.48
SphereFace [106]	2017	A-softmax	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.42
CCL [128]	2018	center invariant loss	ResNet-27	1	CASIA-WebFace (0.49M,10K)	99.12
AMS loss [170]	2018	AMS loss	ResNet-20	1	CASIA-WebFace (0.49M,10K)	99.12
Cosface [172]	2018	cosface	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.33
Arcface [38]	2018	arcface	ResNet-100	1	MS-Celeb-1M (3.8M,85K)	99.83
Ring loss [235]	2018	Ring loss	ResNet-64	1	MS-Celeb-1M (3.5M,31K)	99.50

Ref. M.Wang, "Deep Face Recognition", arXiv:1804.06655

Methodology

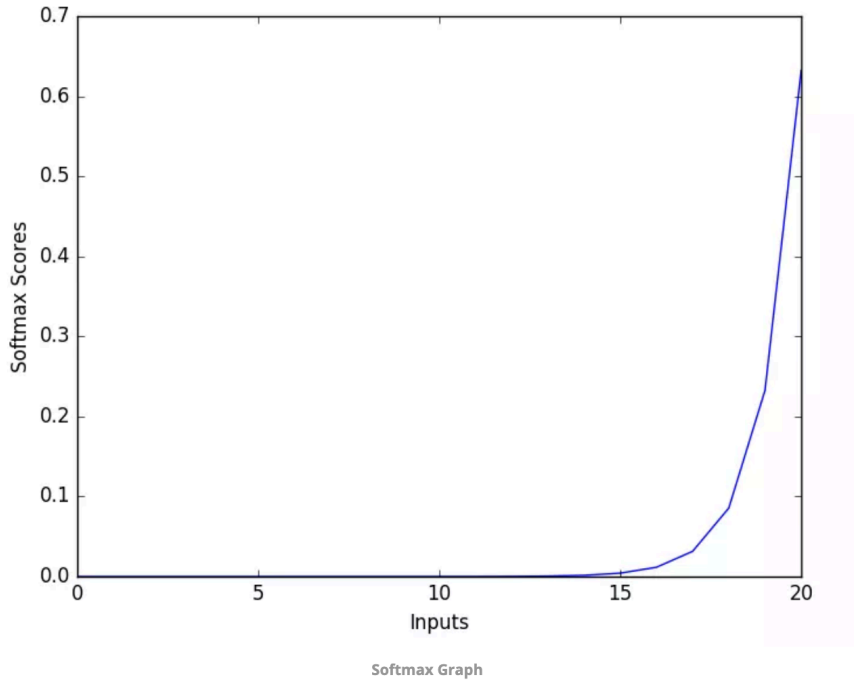
1. Deep Hypersphere Embedding

Revisiting the Softmax

$$p_1 = \frac{\exp(\boldsymbol{W}_1^T \boldsymbol{x} + b_1)}{\exp(\boldsymbol{W}_1^T \boldsymbol{x} + b_1) + \exp(\boldsymbol{W}_2^T \boldsymbol{x} + b_2)}$$

$$p_2 = \frac{\exp(\boldsymbol{W}_2^T \boldsymbol{x} + b_2)}{\exp(\boldsymbol{W}_1^T \boldsymbol{x} + b_1) + \exp(\boldsymbol{W}_2^T \boldsymbol{x} + b_2)}$$

class 1 if $p_1 > p_2$ and class 2 if $p_1 < p_2$.



Softmax Graph

Revisiting the Softmax

$W_1^T x + b_1$ and $W_2^T x + b_2$ determine the classification result.

Decision boundary : $(W_1 - W_2)x + b_1 - b_2 = 0$

Revisiting the Softmax

$W_1^T x + b_1$ and $W_2^T x + b_2$ determine the classification result.

Decision boundary : $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \left\|W_i^T\right\| \|x\| \cos(\theta_i) + b_i$, where θ_i *is the angle between* W_i *and* x

Revisiting the Softmax

$W_1^T x + b_1$ and $W_2^T x + b_2$ determine the classification result.

Decision boundary : $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \left\|W_i^T\right\| \|x\| \cos(\theta_i) + b_i$, where θ_i *is the angle between* W_i *and* x

$normalization$. $(\|W_i\| = 1, b_i = 0)$

Revisiting the Softmax

$W_1^T x + b_1$ and $W_2^T x + b_2$ determine the classification result.

Decision boundary : $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \left\|W_i^T\right\| \|x\| \cos(\theta_i) + b_i$, where θ_i *is the angle between* W_i *and* x

$normalization$. $(\|W_i\| = 1, b_i = 0)$

posterior probabilites : $p_1 = \|x\| \cos(\theta_1)$ *and* $p_2 = \|x\| \cos(\theta_2)$

Revisiting the Softmax

$W_1^T x + b_1$ and $W_2^T x + b_2$ determine the classification result.

Decision boundary : $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$, where θ_i *is the angle between* W_i *and* x

$normalization$. $(\|W_i\| = 1, b_i = 0)$

posterior probabilites : $p_1 = \|x\| \cos(\theta_1)$ *and* $p_2 = \|x\| \cos(\theta_2)$

same x

Revisiting the Softmax

$W_1^T x + b_1$ and $W_2^T x + b_2$ determine the classification result.

Decision boundary : $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$, where θ_i *is the angle between* W_i *and* x

$normalization$. ($\|W_i\| = 1$, $b_i = 0$)

posterior probabilites : $p_1 = \|x\| \cos(\theta_1)$ *and* $p_2 = \|x\| \cos(\theta_2)$

same x

Final result only depends on the angles θ_1 *and* θ_2

Revisiting the Softmax

$W_1^T x + b_1$ and $W_2^T x + b_2$ determine the classification result.

Decision boundary : $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$, where θ_i *is the angle between* W_i *and* x

$normalization$. ($\|W_i\| = 1$, $b_i = 0$)

posterior probabilites : $p_1 = \|x\| \cos(\theta_1)$ *and* $p_2 = \|x\| \cos(\theta_2)$

same x

Final result only depends on the angles θ_1 *and* θ_2

Decision boundary : $\cos(\theta_1) - \cos(\theta_2) = 0$

Revisiting the Softmax

In multi-class case,

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

$$L_i = -\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_j e^{\boldsymbol{W}_j^T \boldsymbol{x}_i + b_j}}\right)$$

$$= -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i}) + b_{y_i}}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_{j,i}) + b_j}}\right)$$

Revisiting the Softmax

In multi-class case,

$$L_i = -\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_j e^{\boldsymbol{W}_j^T \boldsymbol{x}_i + b_j}}\right)$$

$$= -\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i}) + b_{y_i}}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_{j,i}) + b_j}}\right)$$

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

A-Softmax

Formulation, Geometric interpretation

Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

making the decision more stringent

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Decision boundary:

class 1: $\cos(m\theta_1) = \cos(\theta_2)$. ⟶ $\theta_1 < \frac{\theta_2}{m}$

class 2: $\cos(m\theta_2) = \cos(\theta_1)$. ⟶ $\theta_2 < \frac{\theta_1}{m}$

Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Decision boundary:

class 1: $\cos(m\theta_1) = \cos(\theta_2)$. → $\theta_1 < \frac{\theta_2}{m}$

class 2: $\cos(m\theta_2) = \cos(\theta_1)$. → $\theta_2 < \frac{\theta_1}{m}$

A-Softmax

$\theta_1 < \theta_2$

$\theta_2 < \theta_1$

Modified Softmax

Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Decision boundary:

class 1: $\cos(m\theta_1) = \cos(\theta_2)$.

class 2: $\cos(m\theta_2) = \cos(\theta_1)$.

$\theta_1 < \frac{\theta_2}{m}$

$\theta_2 < \frac{\theta_1}{m}$

$\theta_1 < \theta_2$

$\theta_2 < \theta_1$

more difficult (stringent)

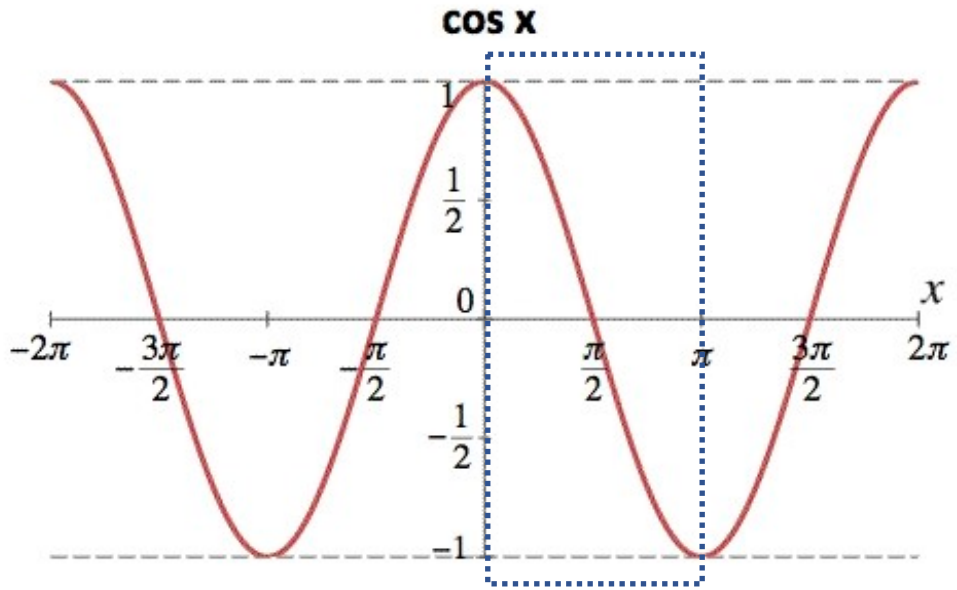
Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

The inequality $\cos(\theta_1) > \cos(m\theta_1)$ holds while $\theta_1 \in [0, \frac{\pi}{m}], m \geq 2$.



Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

The inequality $\cos(\theta_1) > \cos(m\theta_1)$ holds while $\theta_1 \in [0, \frac{\pi}{m}], m \geq 2$.

$\theta_{y_i,i}$ has to be in the range of $[\mathbf{0}, \frac{\boldsymbol{\pi}}{\boldsymbol{m}}]$

Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

The inequality $\cos(\theta_1) > \cos(m\theta_1)$ holds while $\theta_1 \in [0, \frac{\pi}{m}], m \geq 2$.

$\theta_{y_i,i}$ has to be in the range of $[\,\mathbf{0}, \frac{\boldsymbol{\pi}}{\boldsymbol{m}}\,]$

$$\begin{cases} \psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k, \\ \theta_{y_i,i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}] \\ k \in [0, m-1]. \;\; m \geq 1 \end{cases}$$

Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$$\begin{cases} \psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k, \\ \theta_{y_i,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right] \\ k \in [0, m-1]. \;\; m \geq 1 \end{cases}$$

$$L_{\text{ang}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$$\begin{cases} \psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k, \\ \theta_{y_i,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right] \\ k \in [0, m-1]. \quad m \geq 1 \end{cases}$$

$$L_{\text{modified}} = \frac{1}{N}\sum_{i} -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_{j} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$$\begin{cases} \psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k, \\ \theta_{y_i,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right] \\ k \in [0, m-1]. \quad m \geq 1 \end{cases}$$

If m=1, it becomes the modified soft-max loss.

$k \in [0]$
$\theta \in [0, \pi]$
$\psi = \cos(\theta)$

Formulation

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$$\begin{cases} \psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k, \\ \theta_{y_i,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right] \\ k \in [0, m-1]. \quad m \geq 1 \end{cases}$$

If m=1, it becomes the modified soft-max loss.

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

$k \in [0]$
$\theta \in [0, \pi]$
$\psi = \cos(\theta)$

Geometry Interpretation

A-Softmax loss requires $W_i = 1, b_i = 0$,

it makes the prediction **only depends on angels between the sample x and W_i**

Geometry Interpretation

A-Softmax loss requires $W_i = 1, b_i = 0$,

it makes the prediction only depends on angels between the sample x and W_i

x can be classified to the identity with smallest angle.

Geometry Interpretation

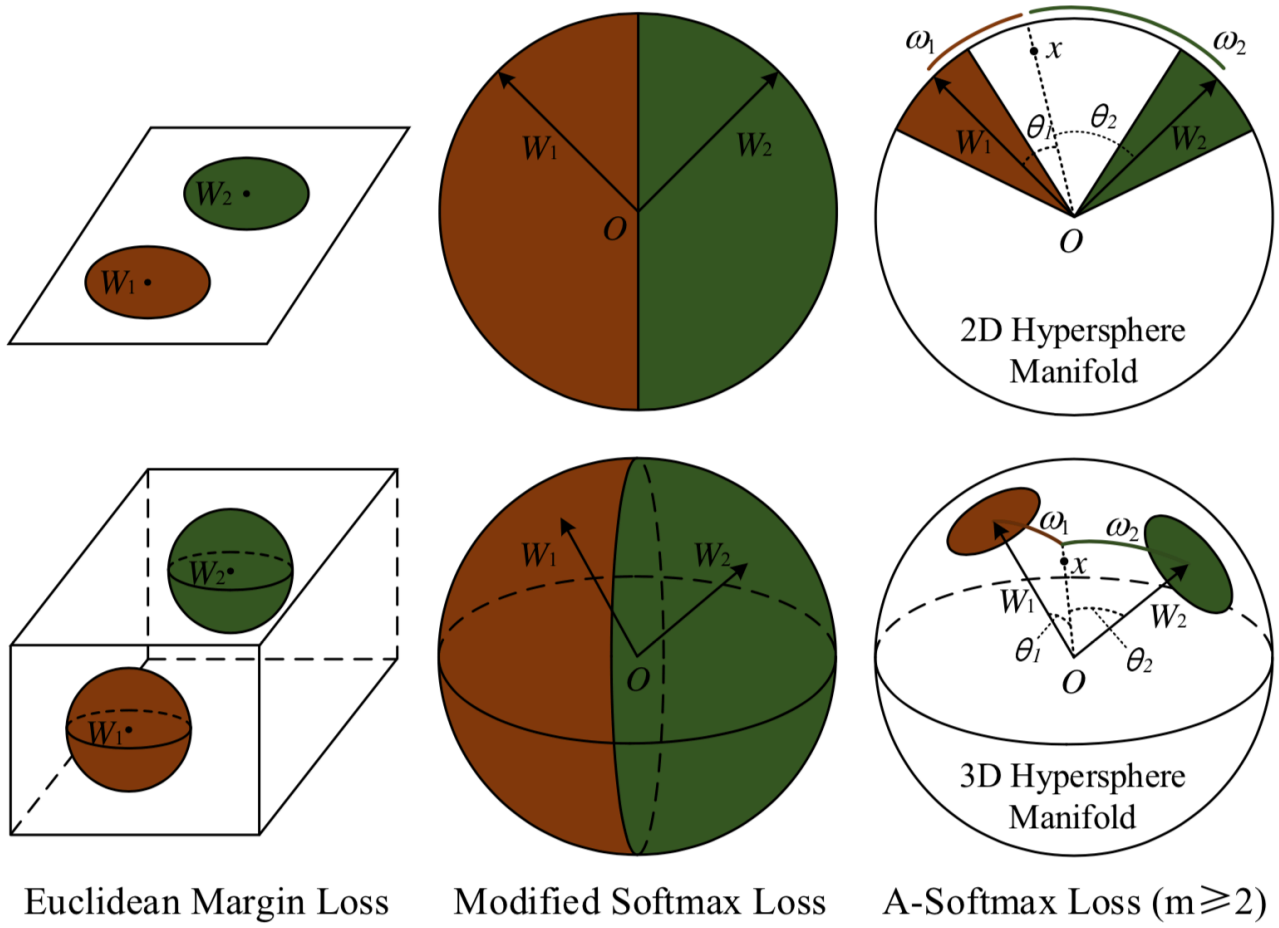
A-Softmax loss requires $W_i = 1, b_i = 0$,

it makes the prediction only depends on angels between the sample x and W_i

x can be classified to the identity with smallest angle.

parameter m is added for the purpose of learning an angular margin between different identities.

Geometry Interpretation



Properties

A-Softmax Loss

Properties of A-Softmax Loss (1/3)

Property 1.

With larger m, the angular margin becomes larger,

Properties of A-Softmax Loss (1/3)

Property 1.

With larger m, the angular margin becomes larger,
the constrained region on the manifold becomes smaller,

Properties of A-Softmax Loss (1/3)

Property 1.

With larger m, the angular margin becomes larger,
the constrained region on the manifold becomes smaller,
and the corresponding learning task also becomes more difficult

Properties of A-Softmax Loss (1/3)

Property 1.

With larger m, the angular margin becomes larger,
the constrained region on the manifold becomes smaller,
and the corresponding learning task also becomes more difficult

Definition 1. (minimal m for desired feature distribution)

m_{min} is the minimal value such that while $m > m_{min}$

maximal intra-class feature distance is constrained to be smaller than the minimal inter-class angular feature distance.

Properties of A-Softmax Loss (2/3)

Property 2. (lower bound of m_{min} in binary-class case)

In binary-class case, we have $m_{min} \geq 2 + \sqrt{3}$

Properties of A-Softmax Loss (2/3)

Property 2. (lower bound of m_{min} in binary-class case)

In binary-class case, we have $m_{min} \geq 2 + \sqrt{3}$

Proof.

We consider the space spaned by W_1 and W_2.
Because $m \geq 2$, maximal angle that class 1 spans is $\frac{\theta_{12}}{m-1} + \frac{\theta_{12}}{m+1}$,
θ_{12} is the angle between W_1 and W_2.

To require the maximal intra-class feature angular distance smaller than the minimal inter-class feature angular distance, we need to constrain

Properties of A-Softmax Loss (2/3)

Proof.

We consider the space spaned by W_1 and W_2.
Because $m \geq 2$, maximal angle that class 1 spans is $\frac{\theta_{12}}{m-1} + \frac{\theta_{12}}{m+1}$,
θ_{12} is the angle between W_1 and W_2.

To require the maximal intra-class feature angular distance smaller than the minimal inter-class feature angular distance, we need to constrain

$$\underbrace{\frac{\theta_{12}}{m-1} + \frac{\theta_{12}}{m+1}}_{\text{max intra-class angle}} \leq \underbrace{\frac{(m-1)\theta_{12}}{m+1}}_{\text{min inter-class angle}}, \quad \theta_{12} \leq \frac{m-1}{m}\pi$$

$$\underbrace{\frac{2\pi - \theta_{12}}{m+1} + \frac{\theta_{12}}{m+1}}_{\text{max intra-class angle}} \leq \underbrace{\frac{(m-1)\theta_{12}}{m+1}}_{\text{min inter-class angle}}, \quad \theta_{12} > \frac{m-1}{m}\pi$$

Properties of A-Softmax Loss (2/3)

Proof.

We consider the space spaned by W_1 and W_2.
Because $m \geq 2$, maximal angle that class 1 spans is $\frac{\theta_{12}}{m-1} + \frac{\theta_{12}}{m+1}$,
θ_{12} is the angle between W_1 and W_2.

To require the maximal intra-class feature angular distance smaller than the minimal inter-class feature angular distance, we need to constrain

$$\underbrace{\frac{\theta_{12}}{m-1} + \frac{\theta_{12}}{m+1}}_{\text{max intra-class angle}} \leq \underbrace{\frac{(m-1)\theta_{12}}{m+1}}_{\text{min inter-class angle}}, \quad \theta_{12} \leq \frac{m-1}{m}\pi$$

$$\underbrace{\frac{2\pi - \theta_{12}}{m+1} + \frac{\theta_{12}}{m+1}}_{\text{max intra-class angle}} \leq \underbrace{\frac{(m-1)\theta_{12}}{m+1}}_{\text{min inter-class angle}}, \quad \theta_{12} > \frac{m-1}{m}\pi$$

$$\Rightarrow \quad m_{min} \geq 2 + \sqrt{3}$$

Properties of A-Softmax Loss (3/3)

Property 3. (lower bound of m_{min} in multi-class case)

Under the assumption that $W_i, \forall i$ are uniformly spaced in the Euclidean space, we have $m_{min} \geq 3$.

Proof.

We consider 2D k-class ($k \geq 3$) scenario for the lower bound. Because $W_i, \forall i$ are uniformly spaced in the 2D Euclidean space, we have $\theta_i^{i+1} = \frac{2\pi}{k}$ where θ_i^{i+1} is the angle between W_i and W_{i+1}.

Properties of A-Softmax Loss (3/3)

Proof.

We consider 2D k-class ($k \geq 3$) scenario for the lower bound. Because $W_i, \forall i$ are uniformly spaced in the 2D Euclidean space, we have $\theta_i^{i+1} = \frac{2\pi}{k}$ where θ_i^{i+1} is the angle between W_i and W_{i+1}.

since $W_i, \forall i$ are symmetric, we only need to analyze one of them. For the i-th class (W_i), We need to constrain

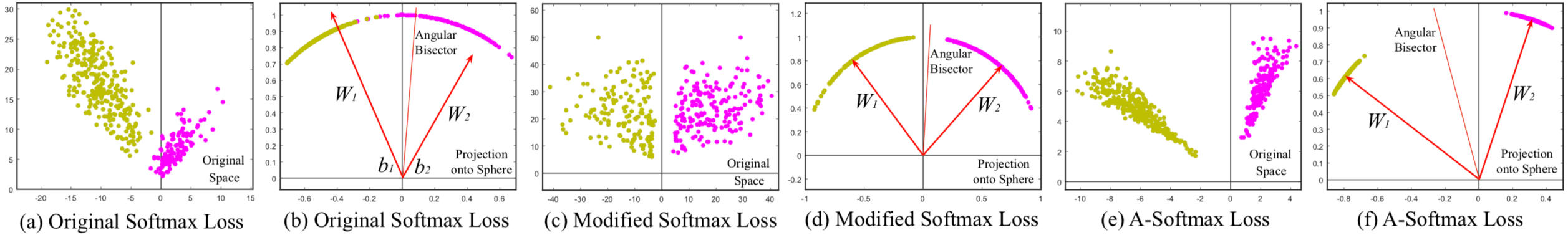
$$\underbrace{\frac{\theta_i^{i+1}}{m+1} + \frac{\theta_{i-1}^{i}}{m+1}}_{\text{max intra-class angle}} \leq \underbrace{\min\left\{\frac{(m-1)\theta_i^{i+1}}{m+1}, \frac{(m-1)\theta_{i-1}^{i}}{m+1}\right\}}_{\text{min inter-class angle}}$$

Discussions

A-Softmax Loss

Why angular margin ?

Angular margin **directly links to discriminativeness on a manifold**, which intrinsically matches the prior that faces also lie on a manifold



(a) Original Softmax Loss (b) Original Softmax Loss (c) Modified Softmax Loss (d) Modified Softmax Loss (e) A-Softmax Loss (f) A-Softmax Loss

Comparison with existing losses.

Contrastive loss, triplet loss and center loss **only impose Euclidean margin**,
while ours instead **directly considers angular margin which is naturally motivated**.

Both contrastive loss and triplet loss **suffer from data expansion when constituting the pairs/triplets from the training set,**
while ours **requires no sample mining** and imposes discriminative constraints to the entire mini-batches.

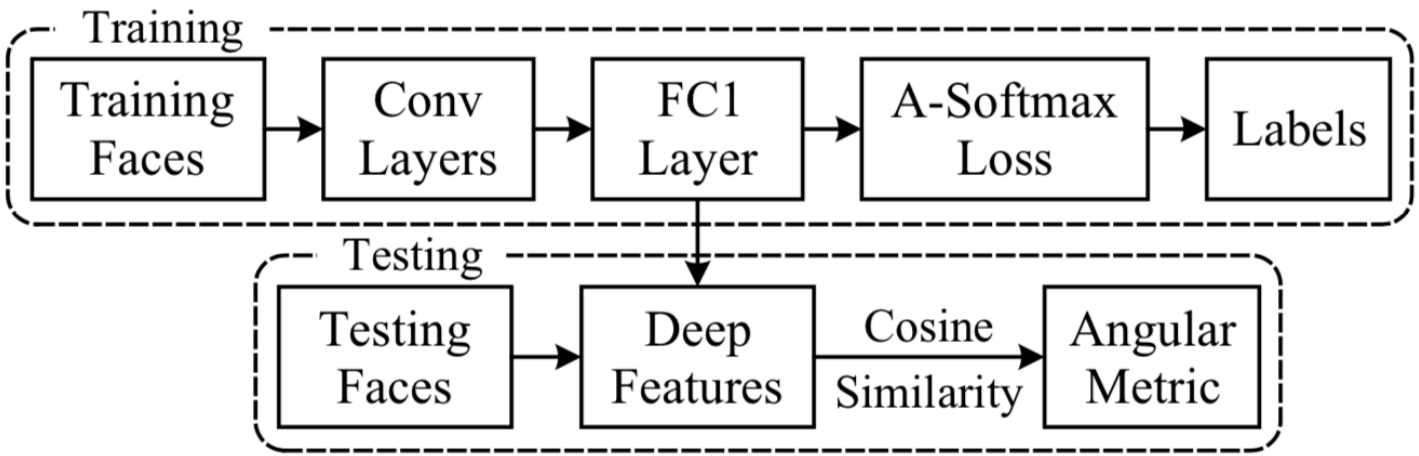
Experiments

Network architecture

Layer	4-layer CNN	10-layer CNN	20-layer CNN	36-layer CNN	64-layer CNN
Conv1.x	[3×3, 64]×1, S2	[3×3, 64]×1, S2	[3×3, 64]×1, S2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times 1$	[3×3, 64]×1, S2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times 2$	[3×3, 64]×1, S2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times 3$
Conv2.x	[3×3, 128]×1, S2	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times 1$	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times 2$	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times 4$	[3×3, 128]×1, S2 $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times 8$
Conv3.x	[3×3, 256]×1, S2	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times 2$	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times 4$	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times 8$	[3×3, 256]×1, S2 $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times 16$
Conv4.x	[3×3, 512]×1, S2	[3×3, 512]×1, S2	[3×3, 512]×1, S2 $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times 1$	[3×3, 512]×1, S2 $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times 2$	[3×3, 512]×1, S2 $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times 3$
FC1	512	512	512	512	512

CNN Setup

- Batch size : 128
- Learning rate : 0.1 → 0.01 (16K iter) → 0.001 (24K iter)
- m : 4
- Training data
  - CASIA-WebFace (494,414 face images)

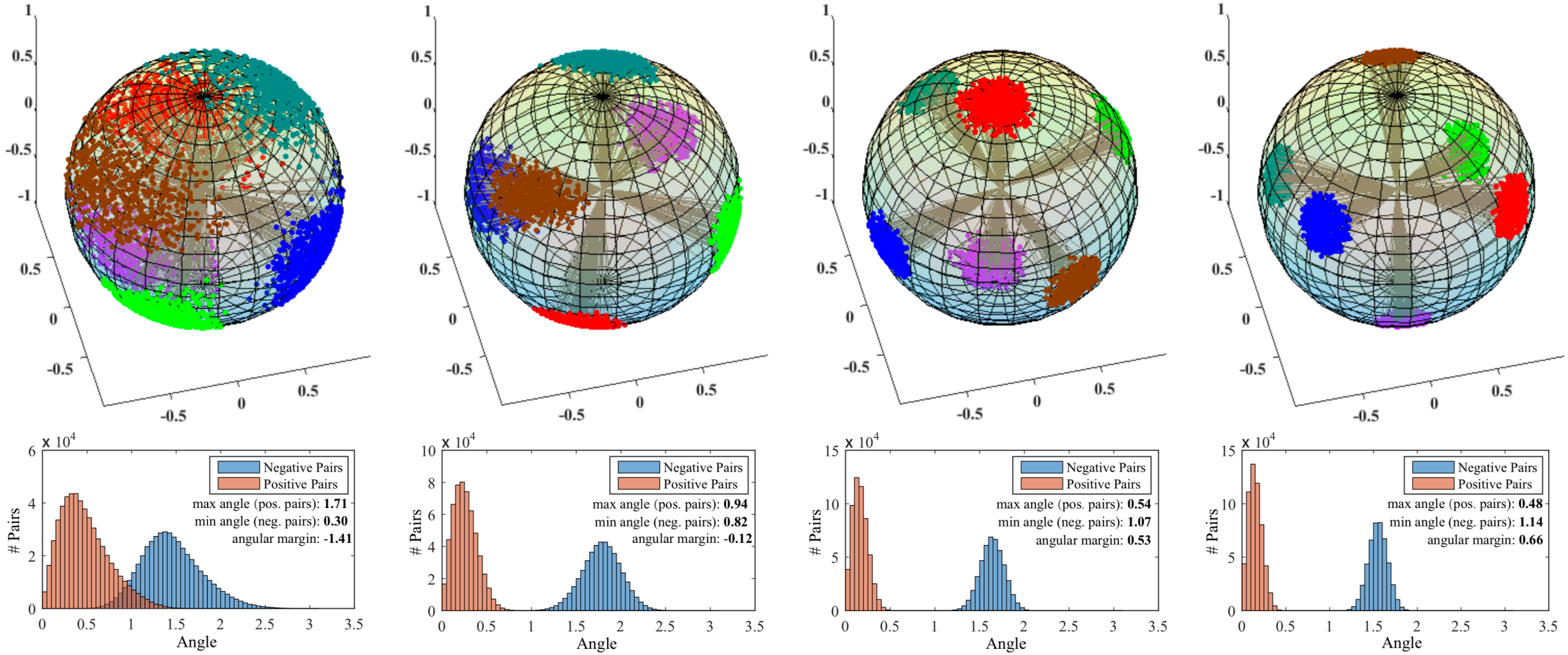


Effect of m

Larger m leads to more discriminative distribution on the sphere.

Dataset	Original	m=1	m=2	m=3	m=4
LFW	97.88	97.90	98.40	99.25	**99.42**
YTF	93.1	93.2	93.8	94.4	**95.0**

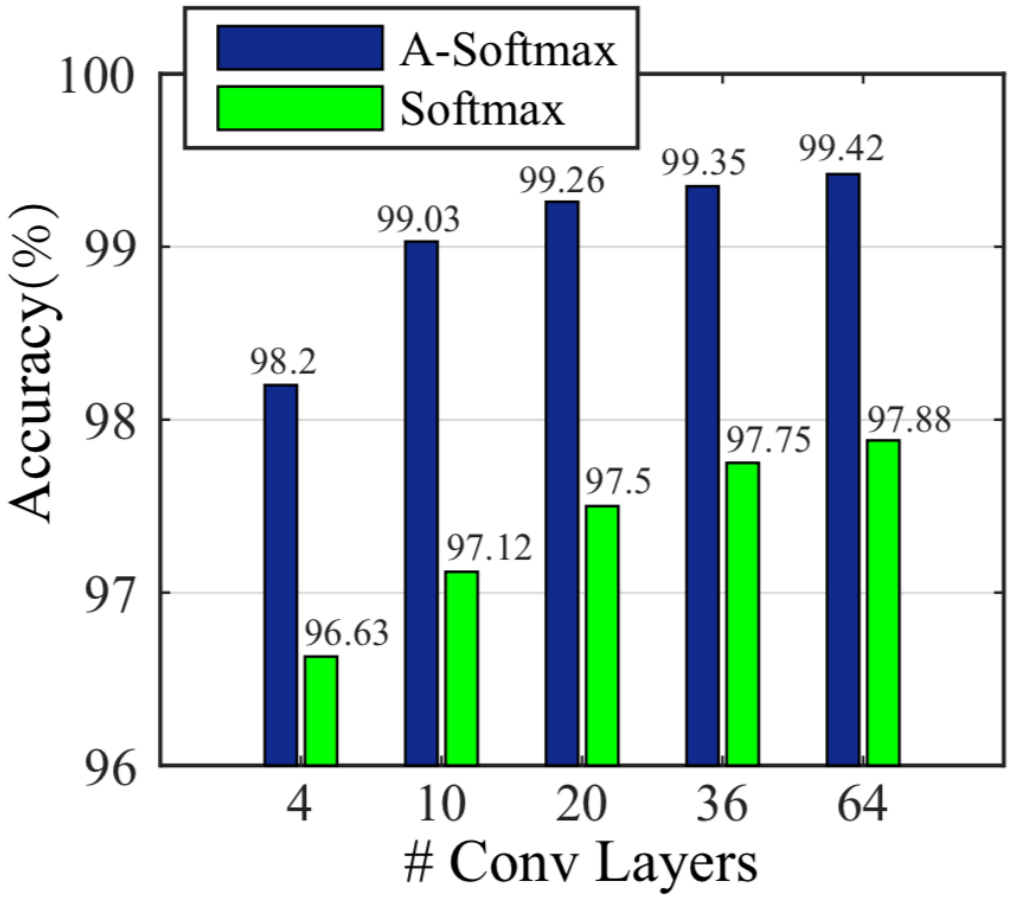
Effect of m



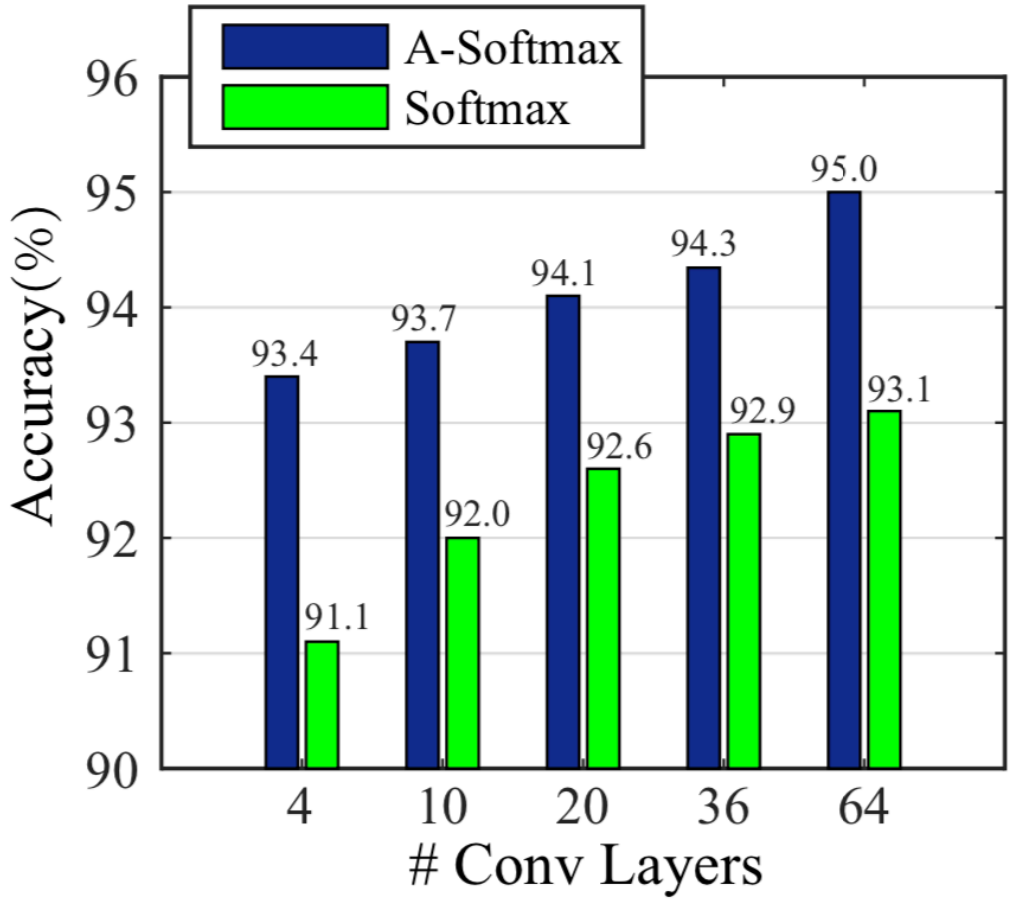
A-Softmax (m=1) A-Softmax (m=2) A-Softmax (m=3) A-Softmax (m=4)

Effect of CNN architectures.

[LFW]



[YTF]



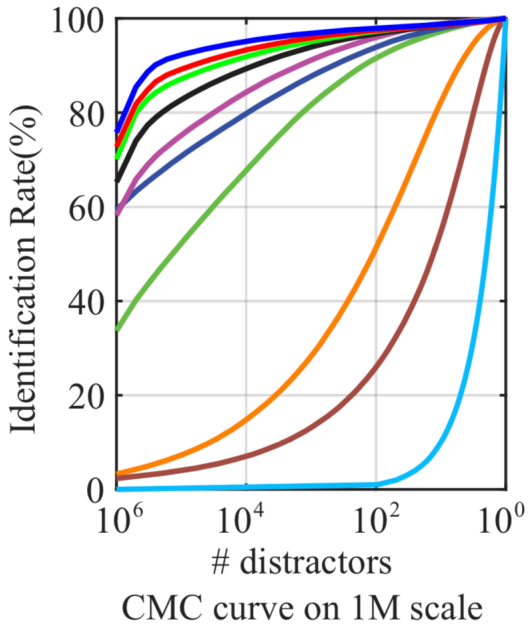
Effect of CNN architectures.

Method	Models	Data	LFW	YTF
DeepFace [30]	3	4M*	97.35	91.4
FaceNet [22]	1	200M*	**99.65**	95.1
Deep FR [20]	1	2.6M	98.95	**97.3**
DeepID2+ [27]	1	300K*	98.70	N/A
DeepID2+ [27]	25	300K*	99.47	93.2
Baidu [15]	1	1.3M*	99.13	N/A
Center Face [34]	1	0.7M*	99.28	94.9
Yi et al. [37]	1	WebFace	97.73	92.2
Ding et al. [2]	1	WebFace	98.43	N/A
Liu et al. [16]	1	WebFace	98.71	N/A
Softmax Loss	1	WebFace	97.88	93.1
Softmax+Contrastive [26]	1	WebFace	98.78	93.5
Triplet Loss [22]	1	WebFace	98.70	93.4
L-Softmax Loss [16]	1	WebFace	99.10	94.0
Softmax+Center Loss [34]	1	WebFace	99.05	94.4
SphereFace	1	WebFace	**99.42**	**95.0**

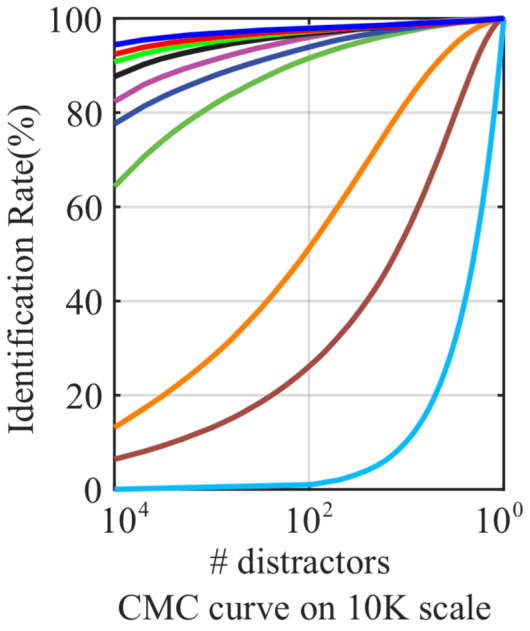
MegaFace Challenge

Method	protocol	Rank1 Acc.	Ver.
NTechLAB - facenx large	Large	73.300	85.081
Vocord - DeepVo1	Large	**75.127**	67.318
Deepsense - Large	Large	74.799	**87.764**
Shanghai Tech	Large	74.049	86.369
Google - FaceNet v8	Large	70.496	86.473
Beijing FaceAll_Norm_1600	Large	64.804	67.118
Beijing FaceAll_1600	Large	63.977	63.960
Deepsense - Small	Small	**70.983**	**82.851**
SIAT_MMLAB	Small	65.233	76.720
Barebones FR - cnn	Small	59.363	59.036
NTechLAB - facenx_small	Small	58.218	66.366
3DiVi Company - tdvm6	Small	33.705	36.927
Softmax Loss	Small	54.855	65.925
Softmax+Contrastive Loss [26]	Small	65.219	78.865
Triplet Loss [22]	Small	64.797	78.322
L-Softmax Loss [16]	Small	67.128	80.423
Softmax+Center Loss [34]	Small	65.494	80.146
SphereFace (single model)	Small	**72.729**	**85.561**
SphereFace (3-patch ensemble)	Small	**75.766**	**89.142**

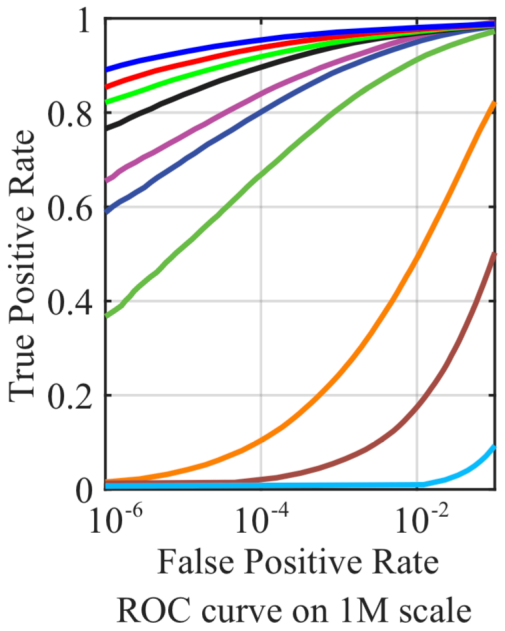
CMC and ROC curves



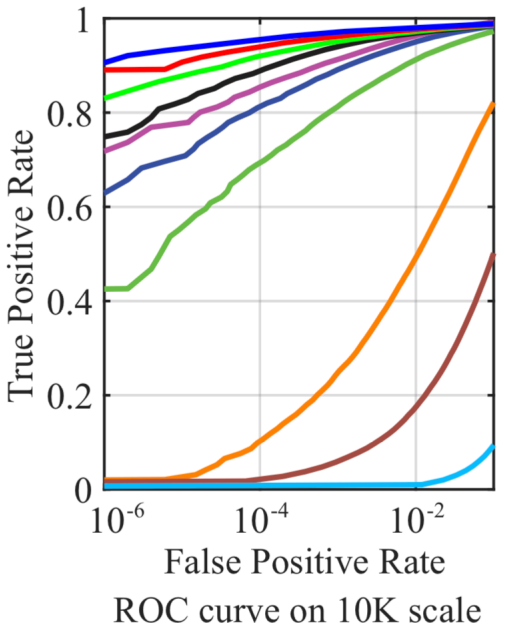
CMC curve on 1M scale



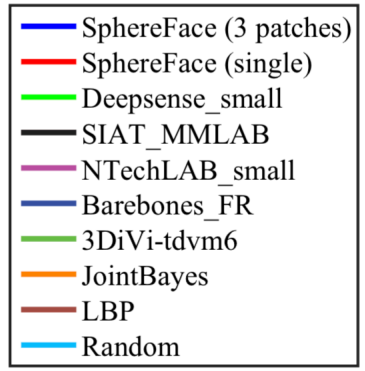
CMC curve on 10K scale



ROC curve on 1M scale



ROC curve on 10K scale

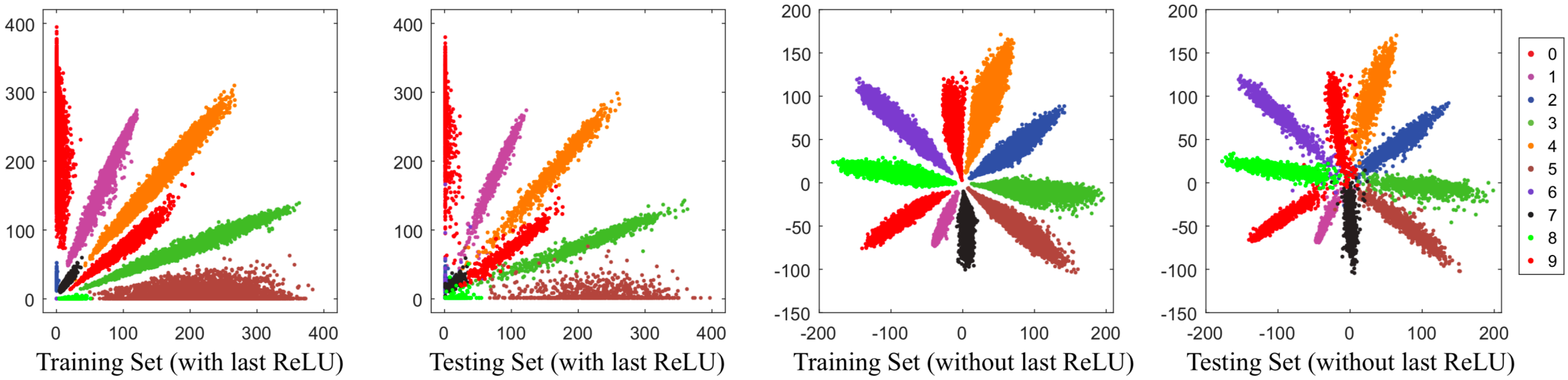


Appendix.

Experiments

Removing the last ReLU

Learned features will only distribute in the non-negative range, which limits the feasible learning space for the CNNs.

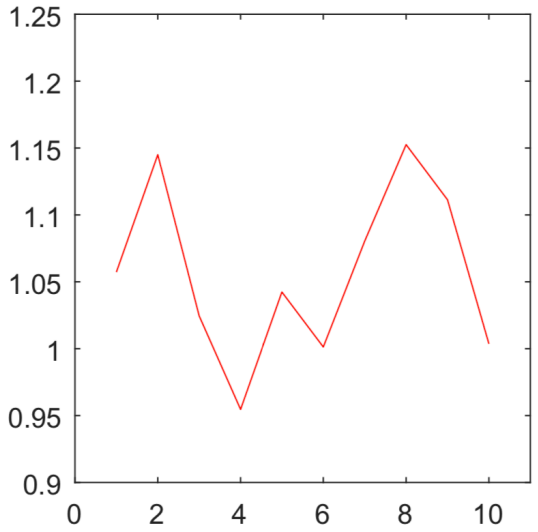


MNIST dataset

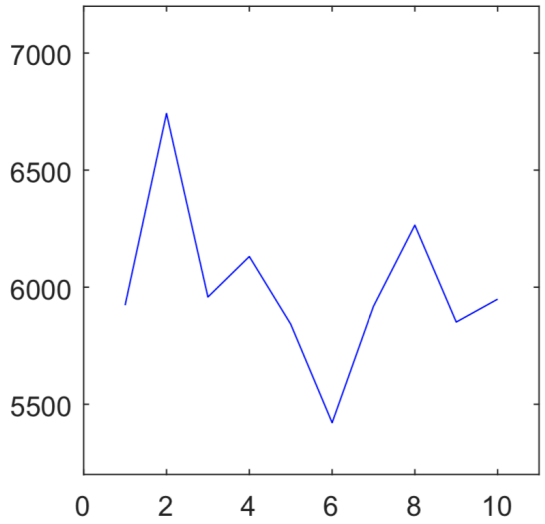
Normalizing the weights

Normalizing the weights can implicitly reduce the prior brought by the training data imbalance issue. (e.g., the long-tail distribution of the training data)

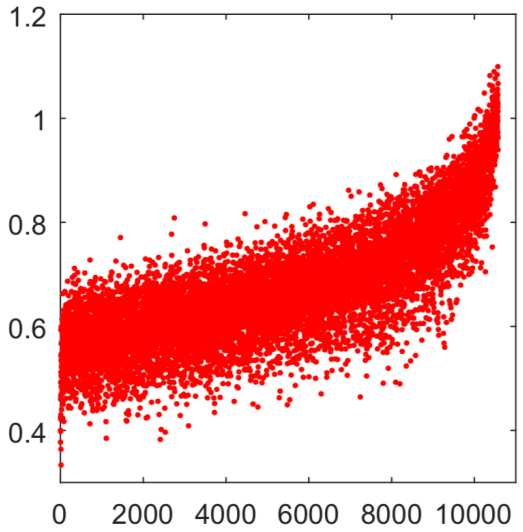
We argue that normalizing the weights can partially address the training data imbalance problem.



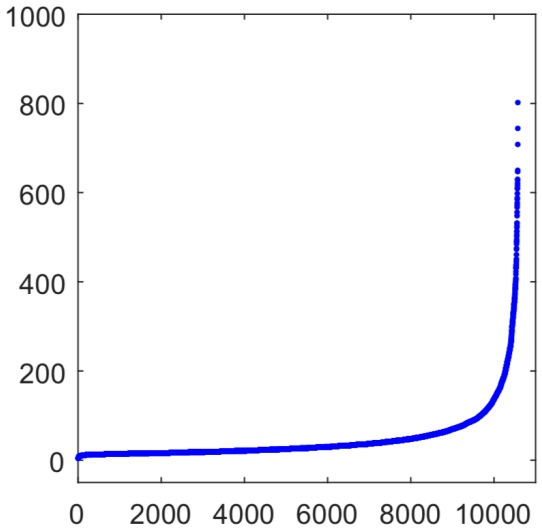
Norm of W_i in MNIST



Sample Number of Class i in MNIST



Norm of W_i in WebFace



Sample Number of Class i in WebFace

Thanks.

☺